Advanced Computer Graphics

Sampling

Matthias Teschner

Computer Science Department
University of Freiburg

Outline

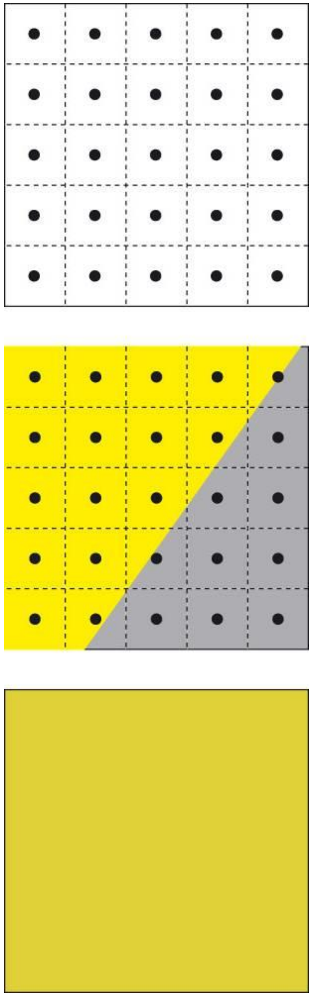
- introduction
- sampling strategies
- low-discrepancy sequences
- mapping samples to a disk, sphere, hemisphere
- reconstruction
- camera effects

Pixel

- radiance at a pixel is reconstructed from discrete point samples of the continuous image function
- there is no area associated with a point sample
- pixel positions are represented with integer values (x, y)
- point samples in the range of [x-0.5, x+0.5) and [y-0.5, y+0.5) contribute to the pixel at position (x, y)
- in implementations, e.g. PBRT, offsets are commonly used, i.e.
  - mapping from continuous c to integer d: $d = \lfloor c \rfloor$
  - mapping from integer d to continuous c: $c = d + 0.5$
  - i.e. [x, x+1) and [y, y+1)
    contribute to the pixel at position (x, y)

Introductory Example

- regular sampling
  - subdivide the pixel area into a regular grid
  - trace a ray per grid cell
- box filter
  - compute the average incident radiance
- effect
  - reduced aliasing due to a higher sampling rate
  - computationally expensive
- goal
  - efficient sampling patterns and filter with reduced aliasing



[Suffern]

Good Sampling Characteristics

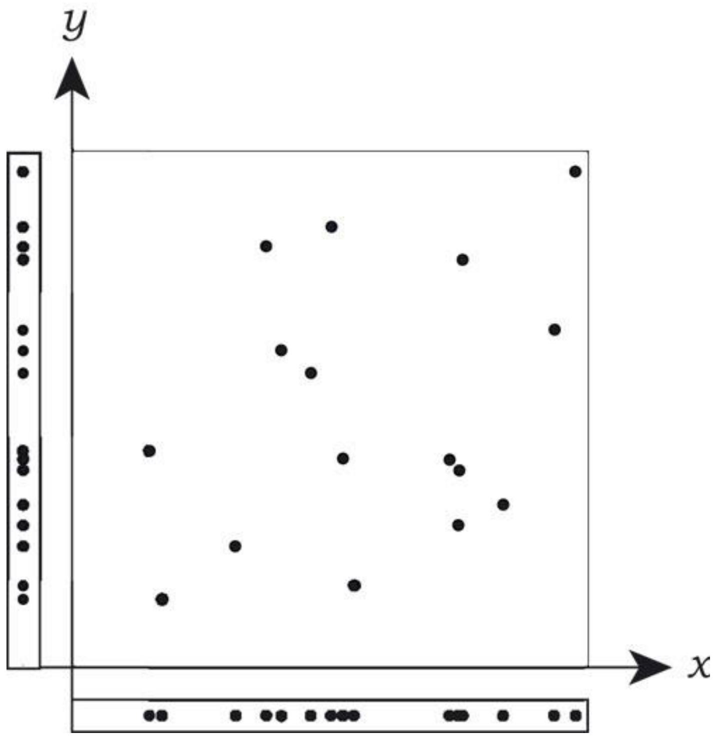
- uniform distribution over the area
- uniform distribution of the projections in x- and y-direction
- maximal minimum distance between samples
  - avoids over- and undersampling of partial areas
- but:
- no regular spacing in x- and / or y-direction
- minimal number of samples with acceptable noise
- number of required samples depends on the application
  - sampling of a pixel area
  - sampling of time for motion blur
  - sampling of the lens area for depth of field
  - sampling of a solid angle for glossy reflection

Outline

- introduction
- sampling strategies
- low-discrepancy sequences
- mapping samples to a disk, sphere, hemisphere
- reconstruction
- camera effects

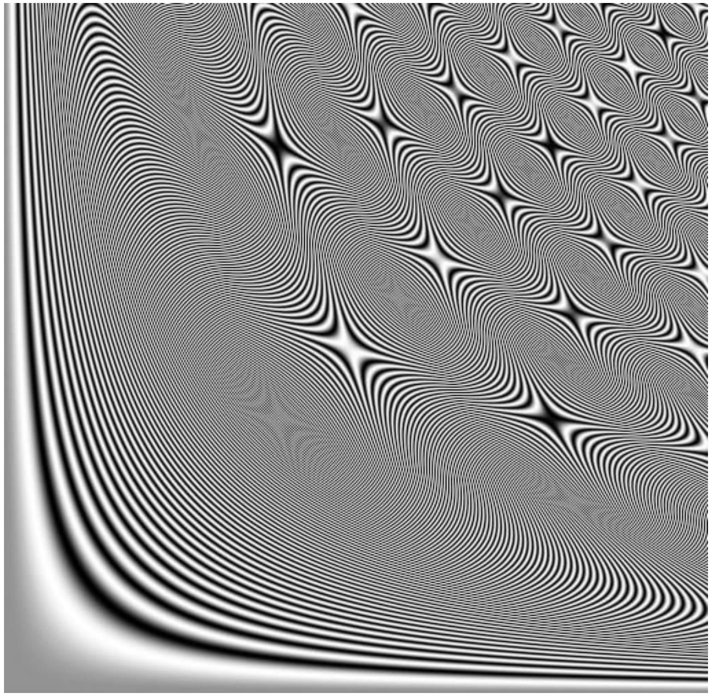
Random Sampling

- replaces aliasing with noise
- requires less samples compared to regular sampling
- non-uniform sampling of partial areas and projections

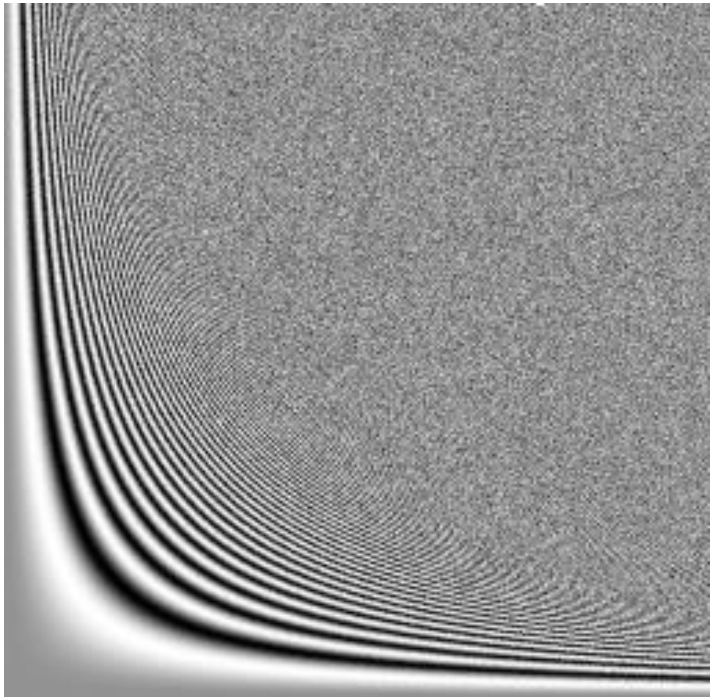


[Suffern]

Random Sampling



regular sampling

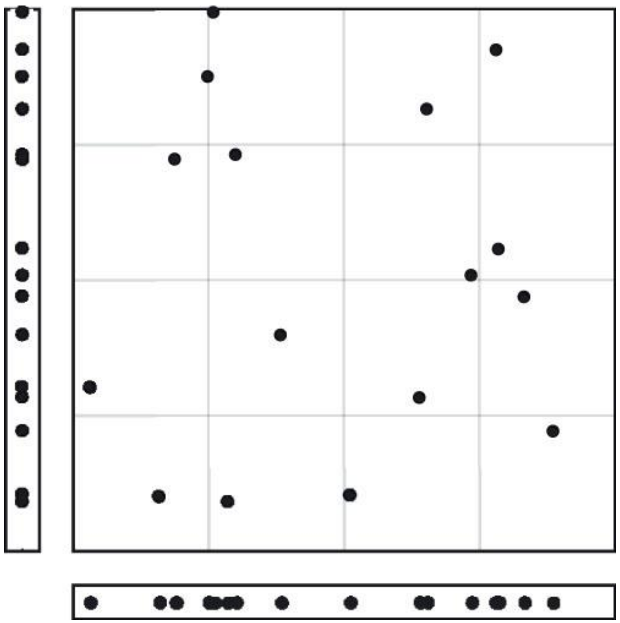


25 random samples

[Suffern]

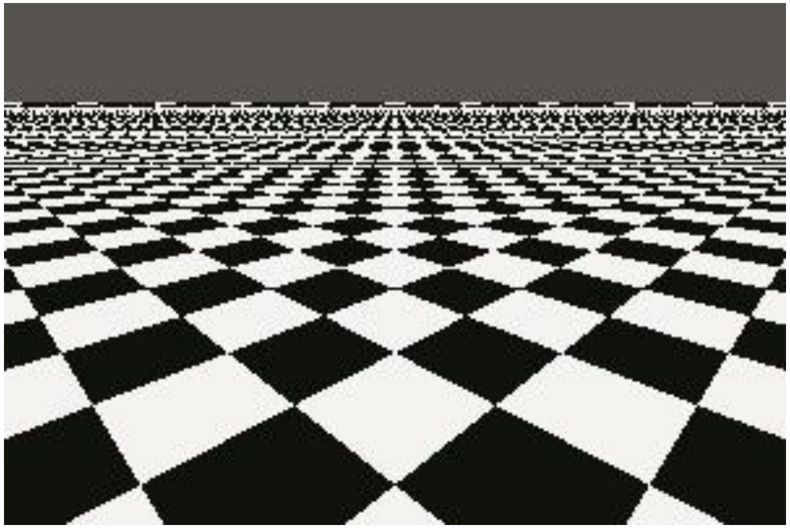
Stratified Sampling

- pixel area is subdivided into n x n strata
- one sample per stratum
- stratified (jittered) sampling reduces clustering of samples, non-uniform sampling of areas and missing of small details

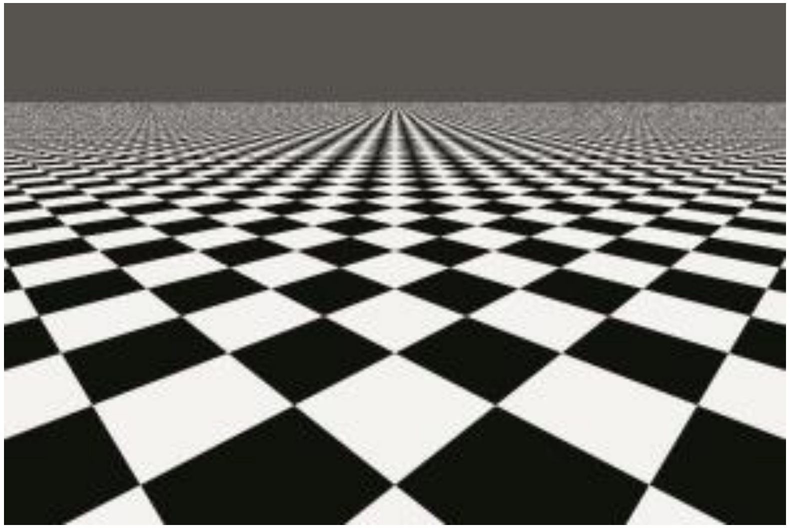


[Suffern]

Stratified Sampling



regular sampling



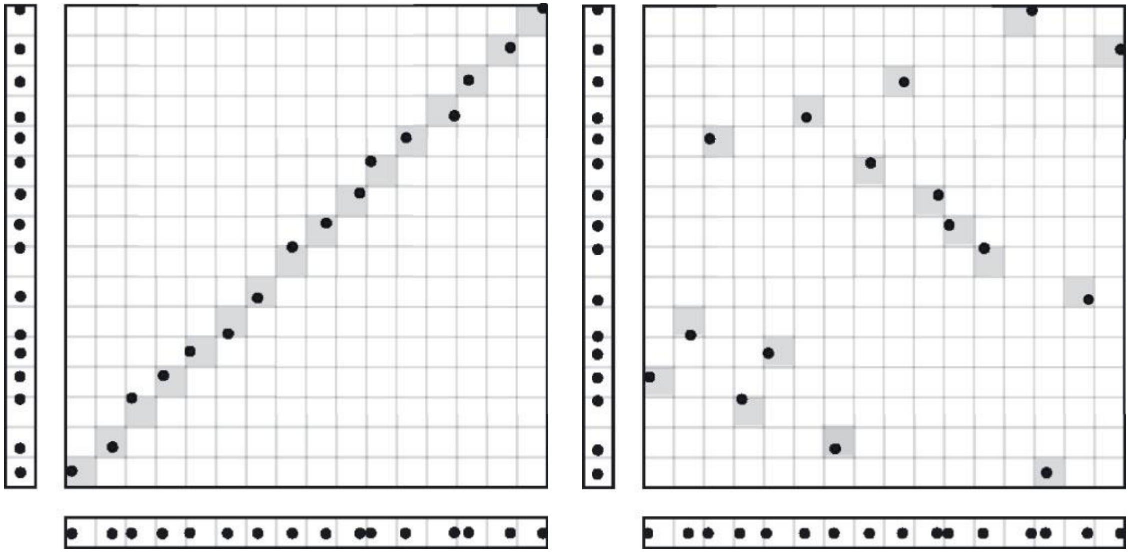
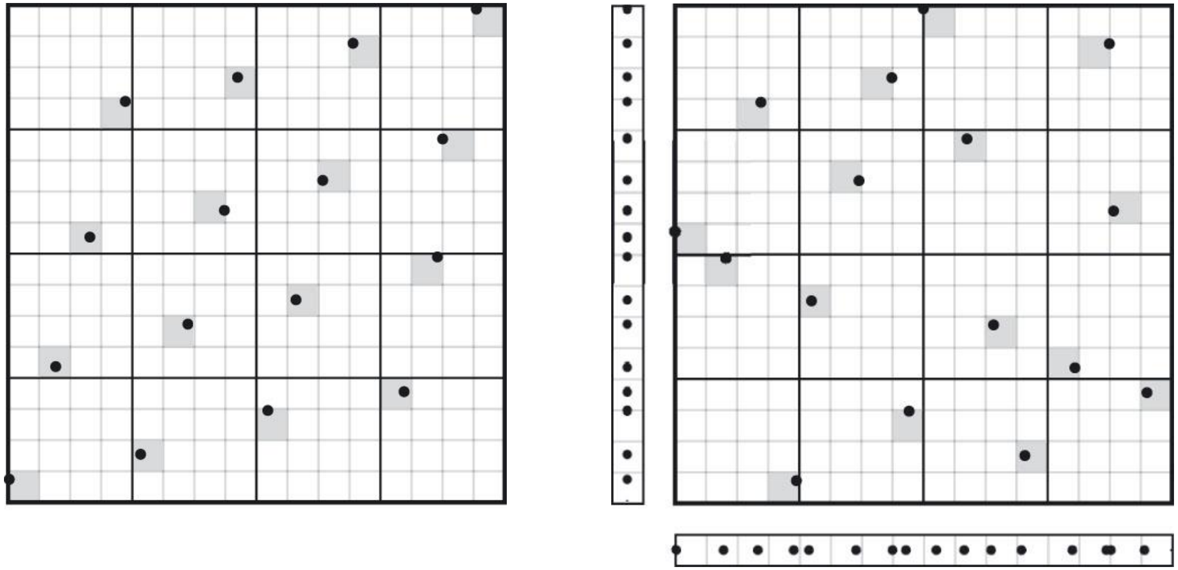
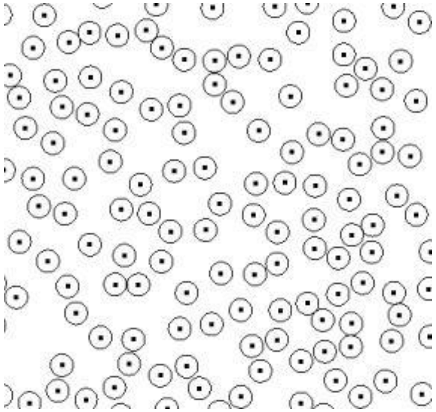
64 stratified jittered samples

[Suffern]

Stratified Sampling in Higher Dimensions

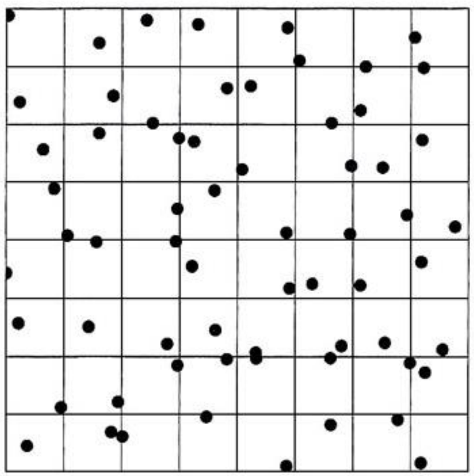
- potentially generates n^d samples with d being the number of degrees of freedom for stratification
  - pixel area $\Rightarrow$ d = 2
  - pixel area + time $\Rightarrow$ d = 3
  - pixel area + time + lens area $\Rightarrow$ d = 5
- instead of generating n^5 samples, only $n^2 + n + n^2$ samples are generated and randomly combined
- similarly, indices in different sample sets can be randomly shuffled
  - to avoid that the same sample combinations are used for different pixels

Half-jittered Sampling

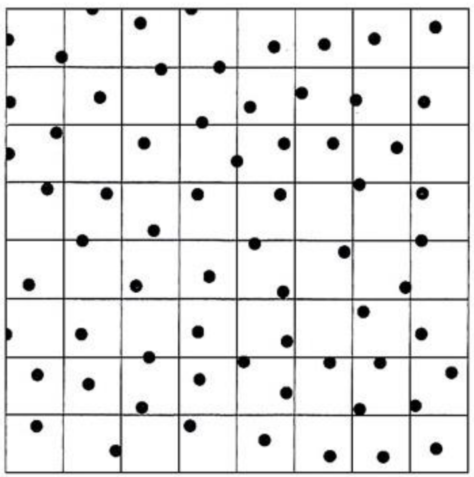
- stratified (jittered) sampling can generate up to four 2D samples at the same position
- solution: choose samples closer to the center of each stratum
  - e.g.
  - ```
    for i=0 to nx-1
        for j=0 to ny-1
            k = i·nx - 1
            xk = randfrom((i+0.25)/nx,(i+0.75)/nx)
            yk = randfrom((j+0.25)/ny,(j+0.75)/ny)
    ```
    ```

n-Rooks / Latin Hypercube Sampling (LHS)

- generate one jittered sample per row and column
- randomly shuffle the samples in x- or y-direction
- uniform distribution of the projections in x- and y-direction
- can generate an arbitrary number of stratified samples



[Suffern]

Multi-jittered Sampling

- initial distribution in a sub grid according to n-rooks
- shuffling in x- and y-direction
- improved distribution over the area
- uniform distribution of the projections in x- and y-direction



[Suffern]

Poisson Disk Sampling

- generate a sequence of random samples
- reject a sample, if it is too close to an existing sample
- rather expensive to compute (dart-throwing)
  - ```
    i = 0
    while i < N
        xi = randfrom [0,1)
        yi = randfrom [0,1)
        reject = false
        for j=0 to i-1
            if (xi - xj)^2 + (yi - yj)^2 < d^2
                reject = true
                break
        if not reject
            i = i+1
    ```



[www.geeks3d.com]
    ```

Best-Candidate Sampling

- generate a larger number of random candidate samples within the entire sampling area
- choose the candidate farthest to previously computed samples



stratified sampling



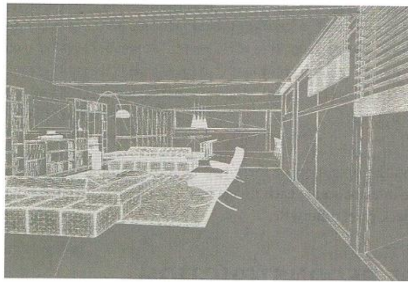
best-candidate sampling

[Pharr, Humphreys]

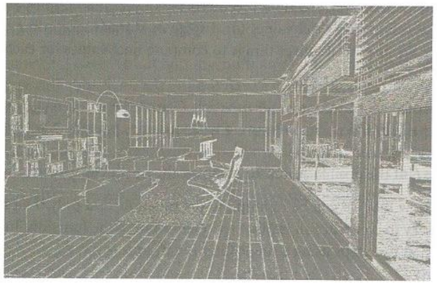
Adaptive Sampling

- generate additional samples per pixel
  - if rays hit more than one shape
  - if radiance values of samples differ significantly

rendered image



adaptive sampling if more than one object intersects the rays of a pixel



adaptive sampling if the radiance per pixel varies significantly

[Pharr, Humphreys]

UNI FREIBURG

Outline

- introduction
- sampling strategies
- low-discrepancy sequences
- mapping samples to a disk, sphere, hemisphere
- reconstruction
- camera effects

Low-Discrepancy Sequences

- good sample sets are characterized by low discrepancy
  - a given fraction of the sampling region, e.g. $[0,1]^d$, should contain the same fraction of sample points
  - difference between the actual region and the region represented by the samples is referred to as discrepancy

$$B = \{[0, v_1] \times [0, v_2] \times \ldots \times [0, v_n]\} \quad 0 \leq v_i \leq 1$$

partial boxes located at the origin

$$P = \{x_1, x_2, \ldots, x_n\}$$

point samples

$$D_N^*(B, P) = \sup_{b \in B} \left| \frac{\sharp\{x_i \in b\}}{N} - \lambda(b) \right|$$

star discrepancy

max — fraction of samples inside the box — fraction of the volume

- low-discrepancy sequence of samples $\Rightarrow$ samples are uniformly distributed

Hammersley Sampling

- non-negative integers *k* can be represented as $k = a_0 + a_1 p + a_2 p^2 + \ldots\ldots + a_r p^r$ with *p* being a prime and integers $a_i \in [0, p-1]$
- $\Phi_p(k) = \frac{a_0}{p} + \frac{a_1}{p^2} + \ldots + \frac{a_r}{p^{r+1}}$ radical inverse function
- $\Phi_2(k)$ for $k = 0, 1, 2, \ldots$ is a Van der Corput sequence
- for primes *p*$_1$, … , *p*$_{d\text{-}1}$, the *k*-th *d*-dimensional Hammersley point of a set with *n* points is

$$\left(\frac{k}{n}, \Phi_{p_1}(k), \Phi_{p_2}(k), \ldots \Phi_{p_{d-1}}(k)\right) \quad k = 0, 1, 2, \ldots, n-1$$

$$p_1 < p_2 < \ldots < p_{d-1}$$

Hammersley Sequence, p=2

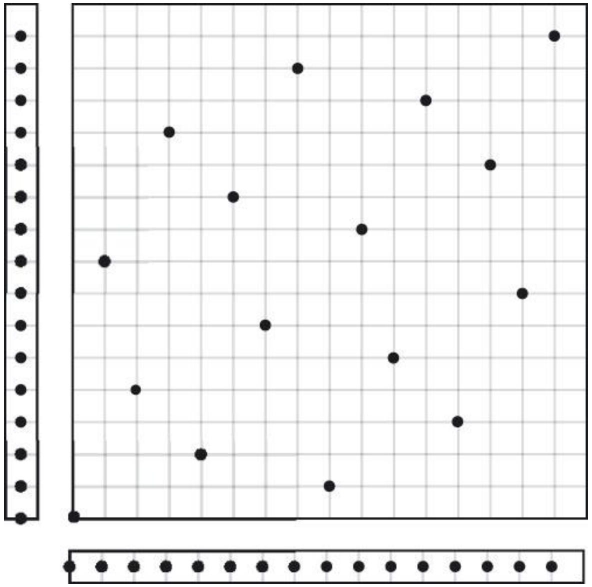
k	binary	binary radical inverse	radical inverse	ϕ_2 (k)
1	1	.1	1/2	0.5
2	10	.01	1/4	0.25
3	11	.11	1/2+1/4	0.75
4	100	.001	1/8	0.125
5	101	.101	1/2+1/8	0.635
6	110	.011	1/4+1/8	0.325
7	111	.111	1/2+1/4+1/8	0.875
8	1000	.0001	1/16	0.0625

$$D_N^* = O\left(\frac{\log N}{N}\right)$$

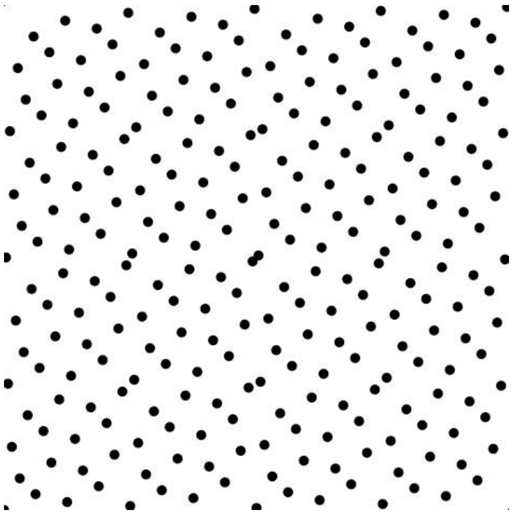
UNI FREIBURG

Hammersley Sampling in 2D

- $\left(\frac{k}{n}, \Phi_2(k)\right) \quad k = 0, 1, 2, \ldots, n-1$



16 samples



256 samples

```
p'=p, k'=k, ϕ=0
while k'>0 do
    a = k' mod p
    ϕ = ϕ + a/p'
    k' = int (k'/p)
    p' = p'*p
```

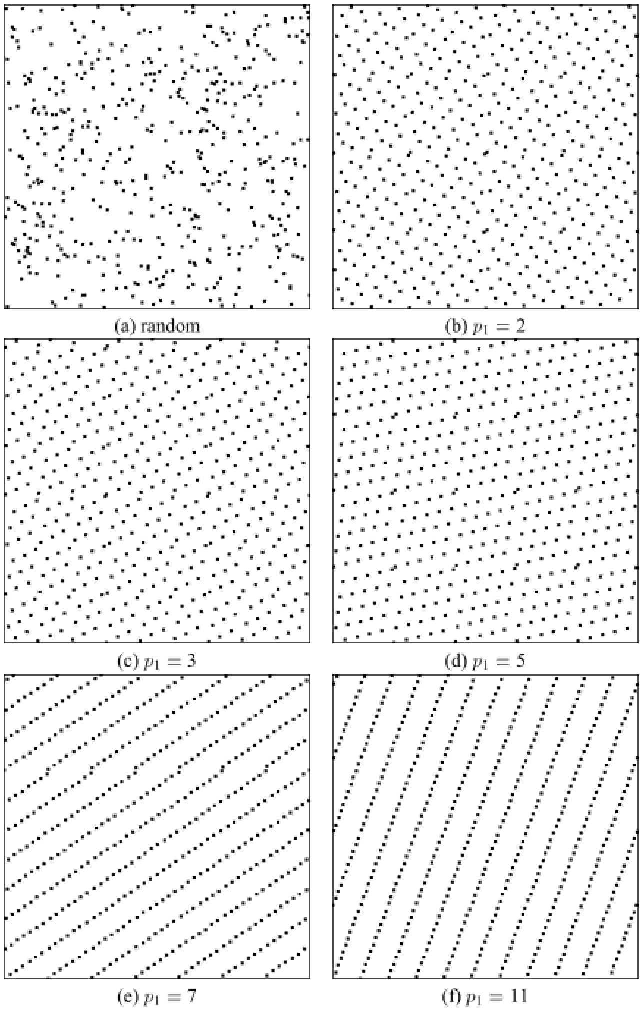
computation of the k-th Hammersley value with basis p

[Suffern]

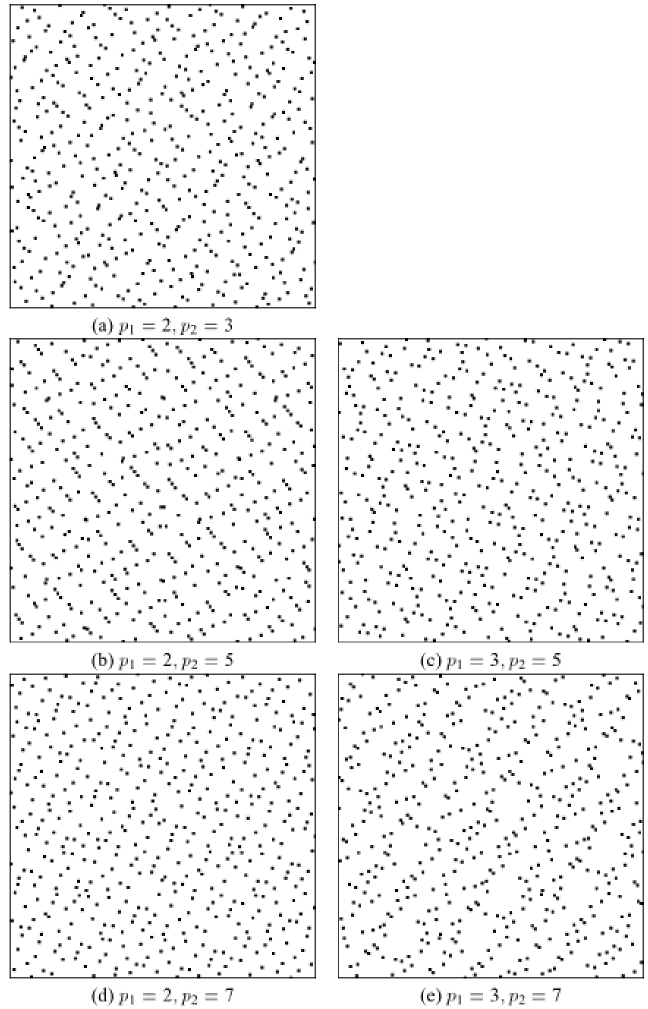
Halton Sampling in 2D

- allows to successively generate additional samples (in contrast to Hammersley)
- $D_N^* = O\left(\frac{(\log N)^d}{N}\right)$
- $(\Phi_{p_1}(k), \Phi_{p_2}(k)) \qquad k = 0, 1, 2, \ldots$

Hammersley vs. Halton - 2D Area



Hammersley sampling



Halton sampling

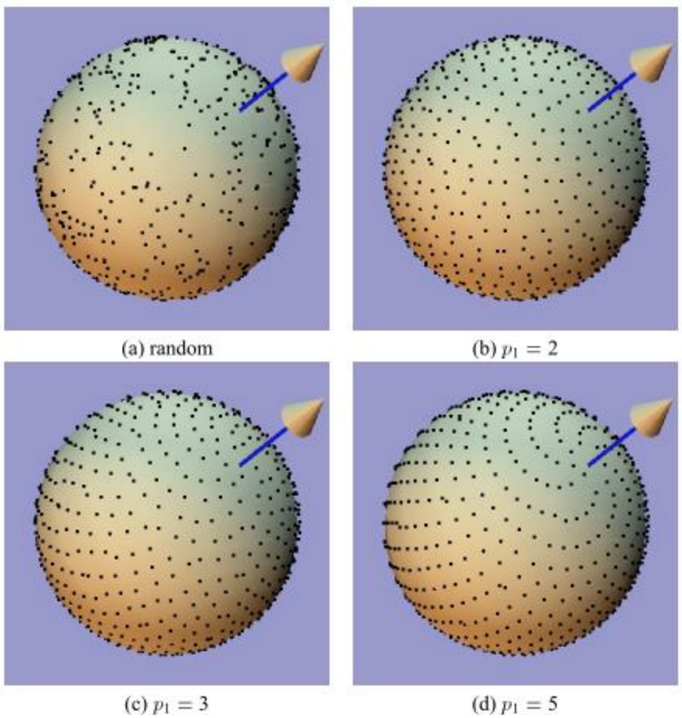
[Wong, Luk, Heng]

Outline

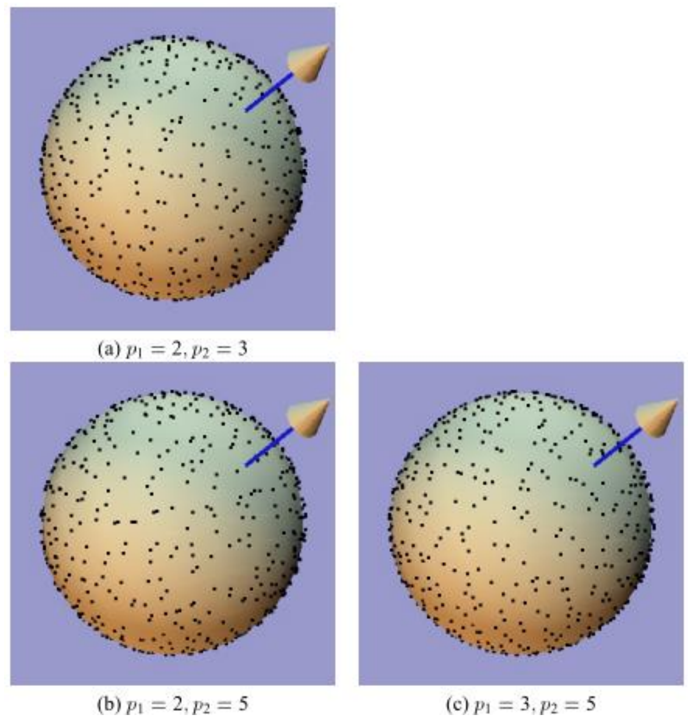
- introduction
- sampling strategies
- low-discrepancy sequences
- mapping samples to a disk, sphere, hemisphere
- reconstruction
- camera effects

Sphere - Hammersley vs. Halton

- $(\frac{k}{n}, \Phi_p(k)) \rightarrow (\phi, t) \in [0, 2\pi) \times [-1, 1]$
  $$\rightarrow (\sqrt{1-t^2}\cos\phi, \sqrt{1-t^2}\sin\phi, t)$$



Hammersley sampling

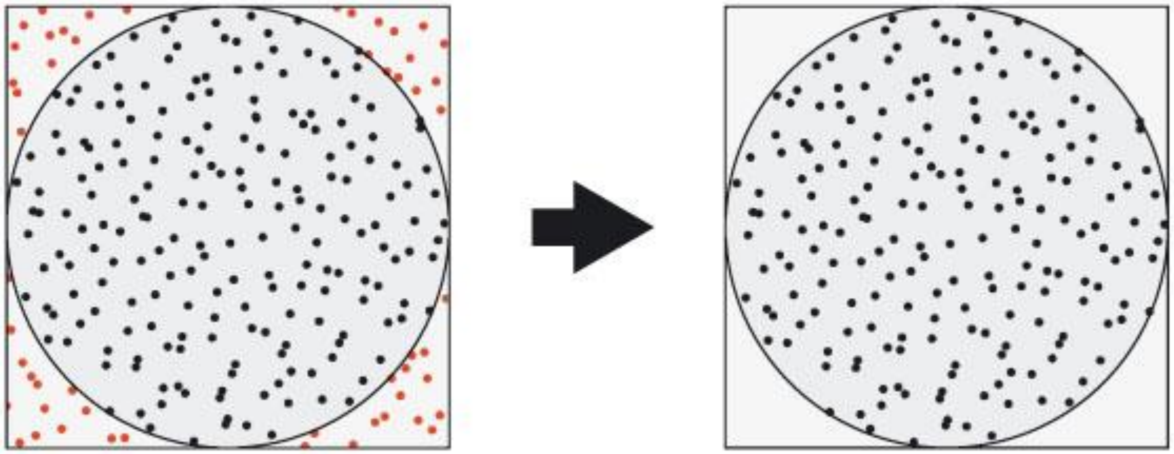


Halton sampling

[Wong, Luk, Heng]

Disk - Rejection Sampling

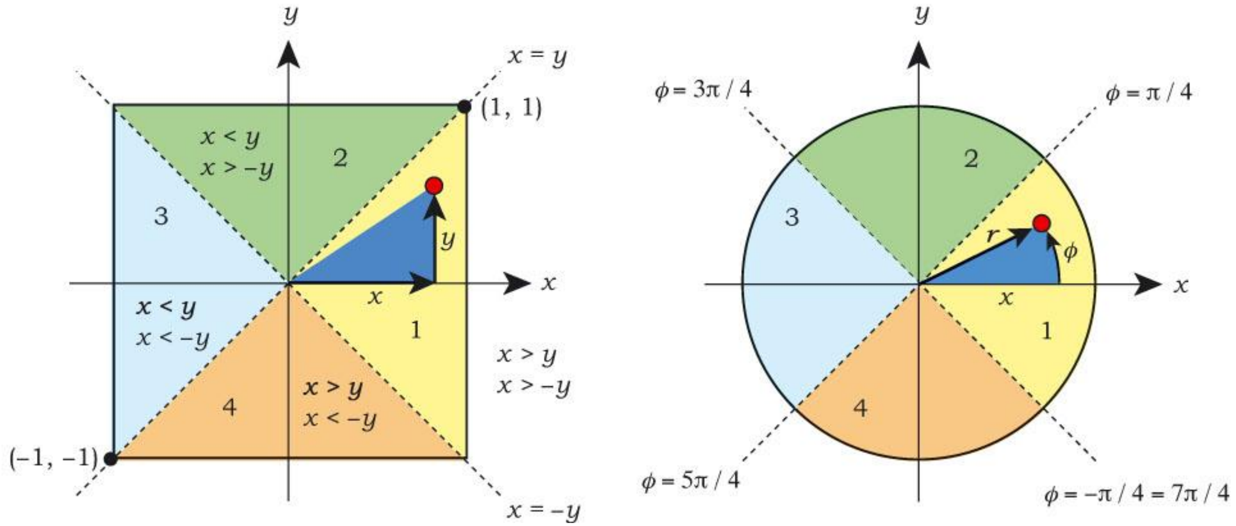
- sample points outside a disk are rejected
- no distortion
- but differing number of remaining samples



[Suffern]

Disk - Concentric Map

- mapping from square to disk
- minimal distortion
- number of samples is preserved

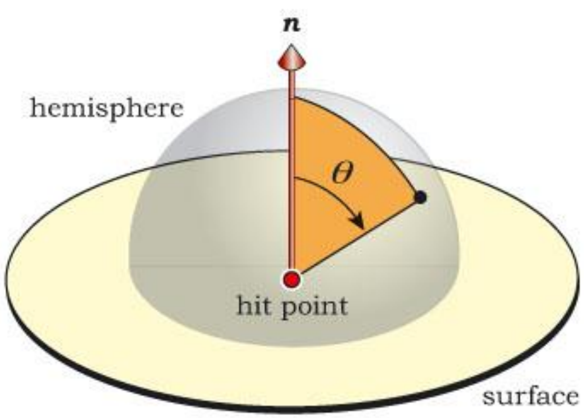


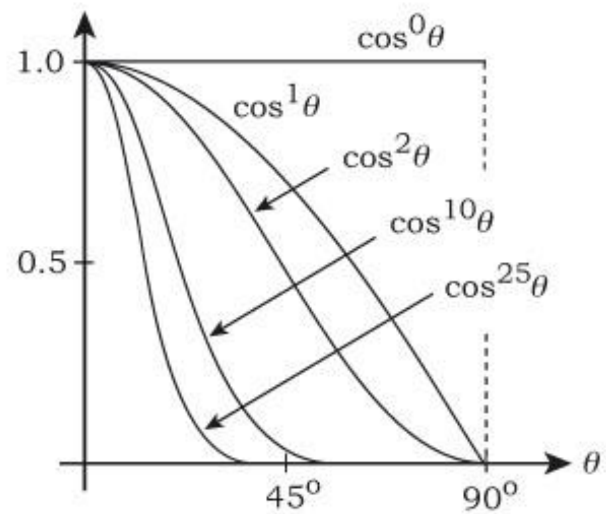
- e.g., $(x > y, x > -y) \to (r = x, \phi = \frac{\pi}{4}\frac{y}{x})$
- four quarters

[Suffern]

Hemisphere

- mapping from square to hemisphere with a cosine power density distribution
- surface density of samples varies with θ according to $d = \cos^m \theta$



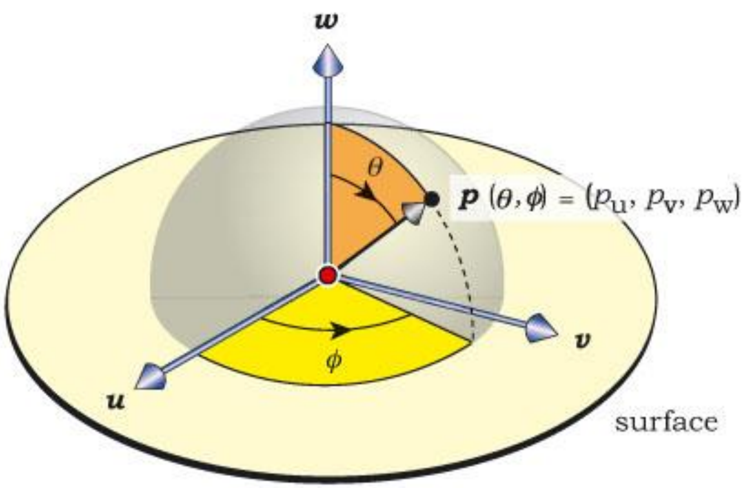


[Suffern]

Hemisphere

- mapping:

$$(x \in [0,1), y \in [0,1)) \rightarrow (\phi = 2\pi x, \theta = \cos^{-1}[(1-y)^{1/(m+1)}])$$

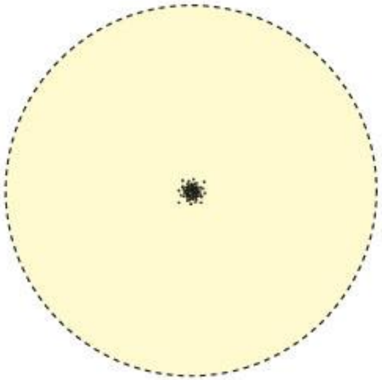


local orthonormal basis at the center of the hemisphere

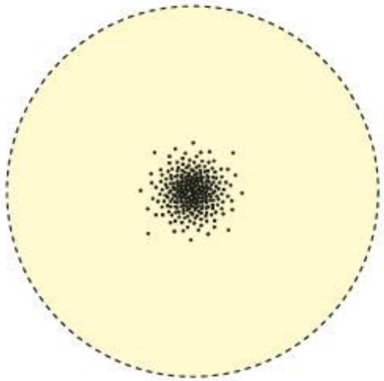
$$\mathbf{p} = \sin\theta\cos\phi\,\mathbf{u} + \sin\theta\sin\phi\,\mathbf{v} + \cos\theta\,\mathbf{w}$$

[Suffern]

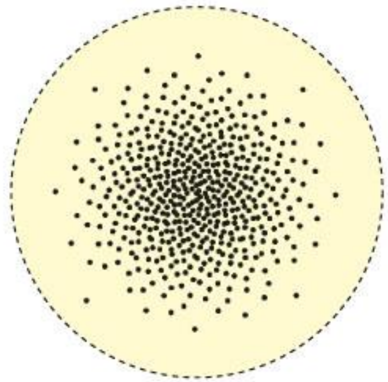
Hemisphere Sampling



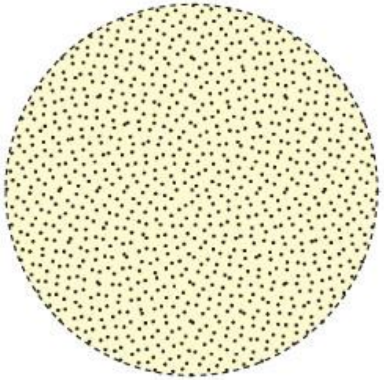
$m = 1000$



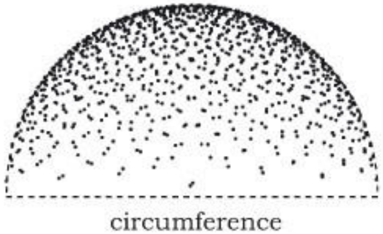
$m = 100$



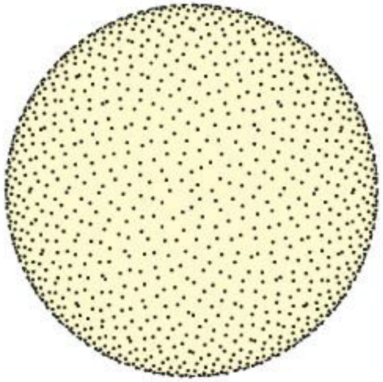
$m = 10$



$m = 1$



$m = 1$



$m = 0$

[Suffern]

Applications

- area sampling is used to sample pixel areas
- hemisphere sampling is used for global illumination effects
- disk sampling is used for the depth-of-field effect

Outline

- introduction
- sampling strategies
- low-discrepancy sequences
- mapping samples to a disk, sphere, hemisphere
- reconstruction
- camera effects

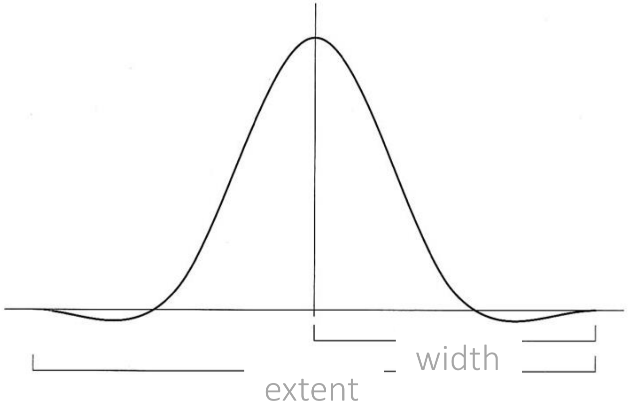
Filtering Principle

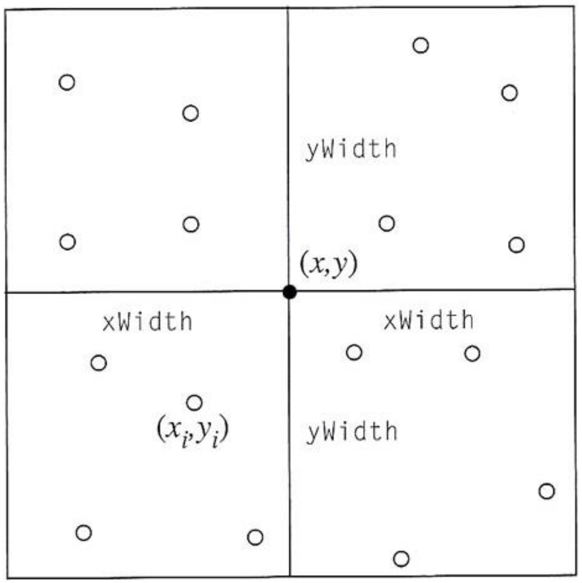
- pixel values are reconstructed from the radiance values of adjacent samples

$$\mathbf{I}(x,y) = \frac{\sum_i f(x-x_i, y-y_i)\mathbf{L}(x_i,y_i)}{\sum_i f(x-x_i, y-y_i)}$$

- f is a filter function that weights the influence of a sample (x_i, y_i) according to its distance to (x, y)

the extent can be larger than the pixel size



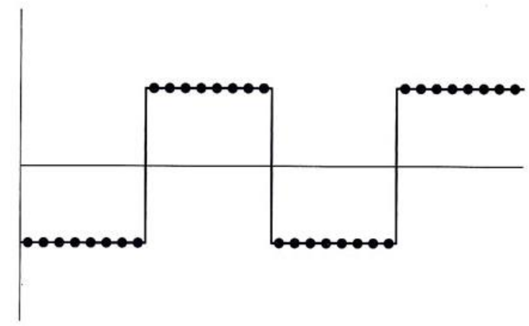


[Pharr, Humphreys]

Box- / Triangle Filter

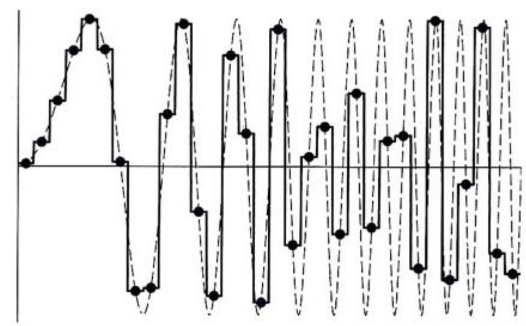
- computationally efficient
- bad reconstruction characteristics

box filter applied to a step function

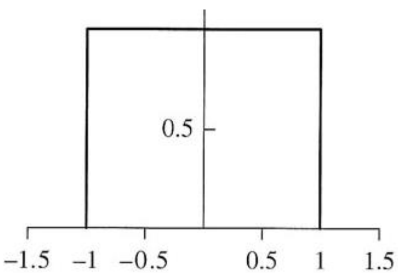


(a)

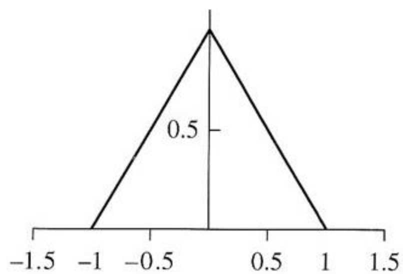
box filter applied to a sinusoidal function with in-icreasing frequency ⇒ introduces post-aliasing



(b)



box filter



triangle filter

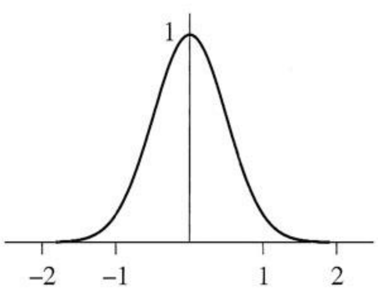
[Pharr, Humphreys]

Gaussian- / Mitchell Filter

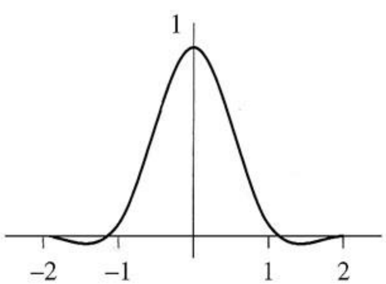
- reasonably good reconstruction
- introduces blurring
- e.g. , 1D filter

$$f(x) = e^{-\alpha x^2} - e^{-\alpha w^2}$$

offset according
to the filter width w

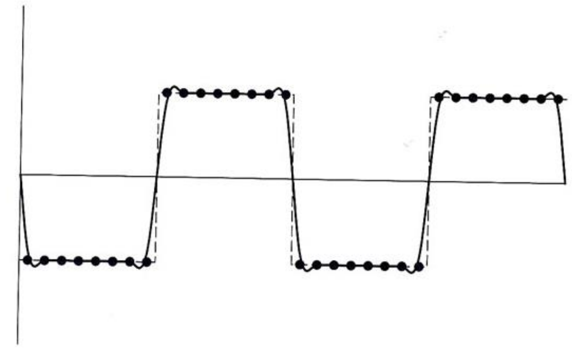


Gaussian filter



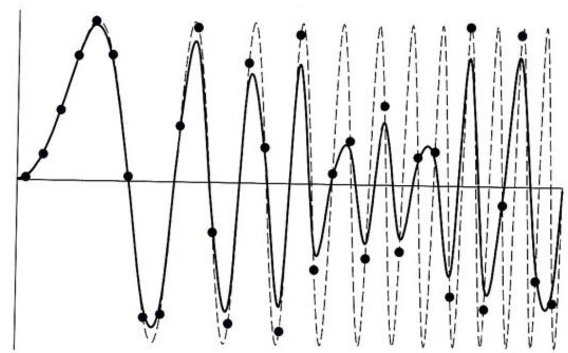
Mitchell filter

Mitchell filter applied
to a step function
(minimal ringing)



(a)

Mitchell filter applied
to a sinusoidal
function with in-
icreasing frequency
⇒ more aliasing
compared to Mitchell
due to undersampling

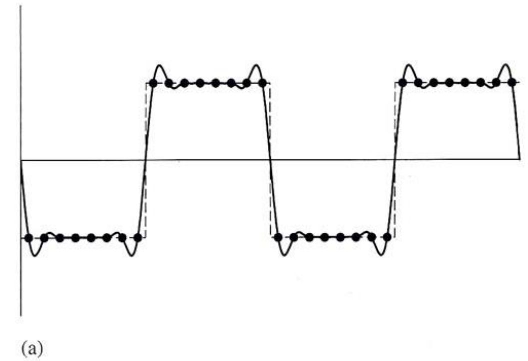


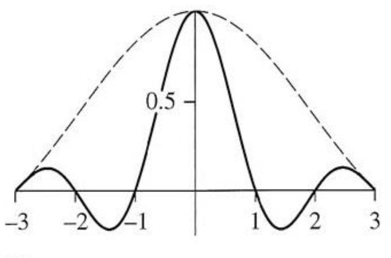
[Pharr, Humphreys]

Truncated Sinc Filter

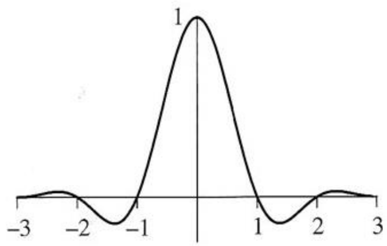
- reasonably good reconstruction
- introduces blurring

Sinc filter applied to a step function (some ringing)



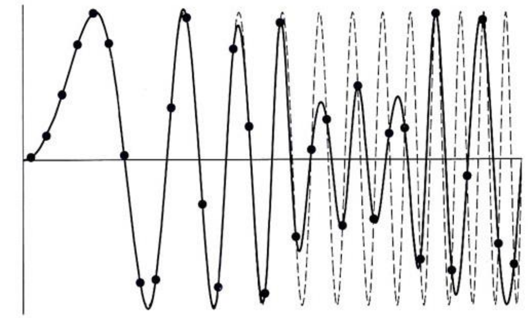


sinc filter



Lanczos sinc filter (sinc times windowing function)

Sinc filter applied to a sinusoidal function with in-icreasing frequency ⇒ aliasing due to undersampling



[Pharr, Humphreys]

UNI FREIBURG

Summary

- sampling strategies for pixel area, disk, hemisphere help to reduce aliasing or to replace it with noise
  - stratified sampling
  - Poisson disk sampling
  - low discrepancy sequences
  - mapping to disk, hemisphere, sphere
- reconstruction of pixel values from sample radiances
  - box
  - Mitchell
  - truncated sinc

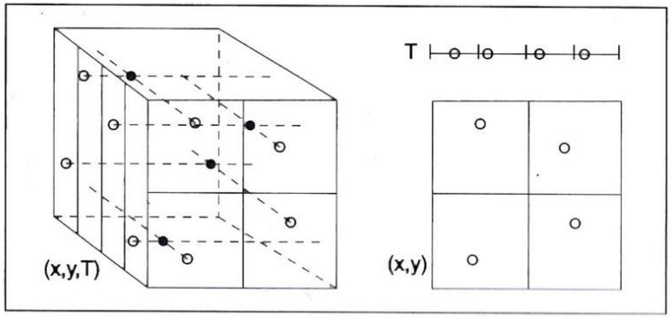
Outline

- introduction
- sampling strategies
- low-discrepancy sequences
- mapping samples to a disk, sphere, hemisphere
- reconstruction
- camera effects

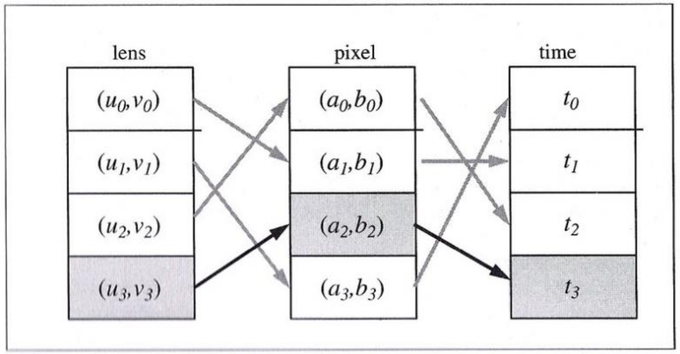
UNI FREIBURG

Multidimensional Sampling

- sampling the pixel area
  - captures the continuous radiance function
- sampling a time period
  - captures motion blur effects
- sampling a disk
  - captures depth-of-field effects



sampling of pixel and time

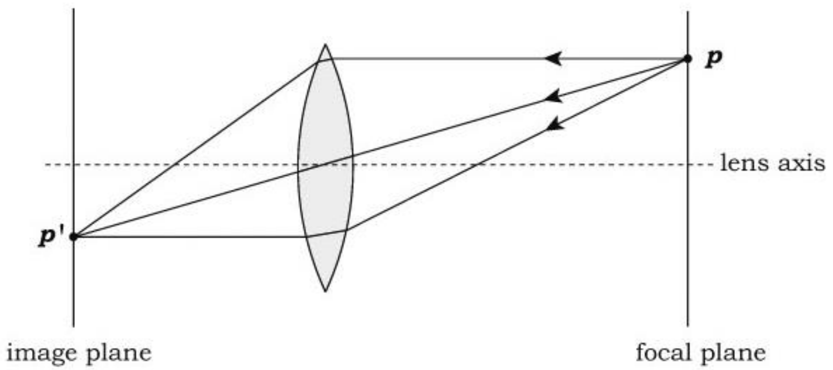


sampling of lens, pixel and time

[Shirley, Morley]

Thin Lens

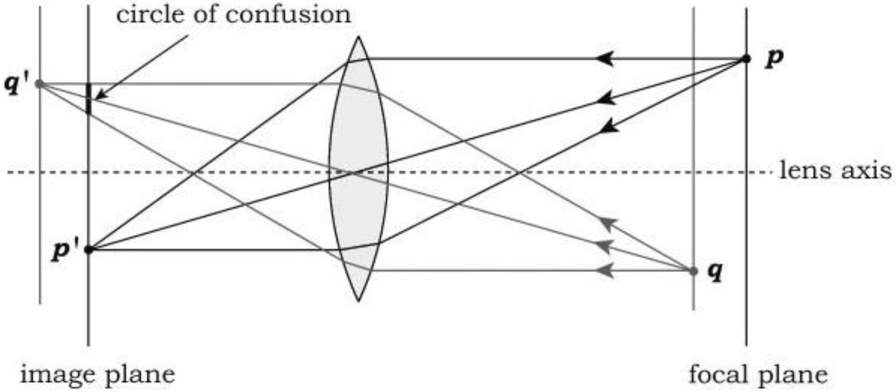
- instead of a pinhole camera, we model a thin lense
  - every point **p** on the focal plane has a corresponding point **p'** on the image plane
  - every ray that goes through **p** and the lens also goes through **p'**
  - a ray through the center of the lens is not refracted



[Suffern]

Circle of Confusion

- **q** is not on the focal plane
- rays that go through **q** and the lens do not intersect at a point on the image plane
- instead, the intersections with the image plane form the circle of confusion



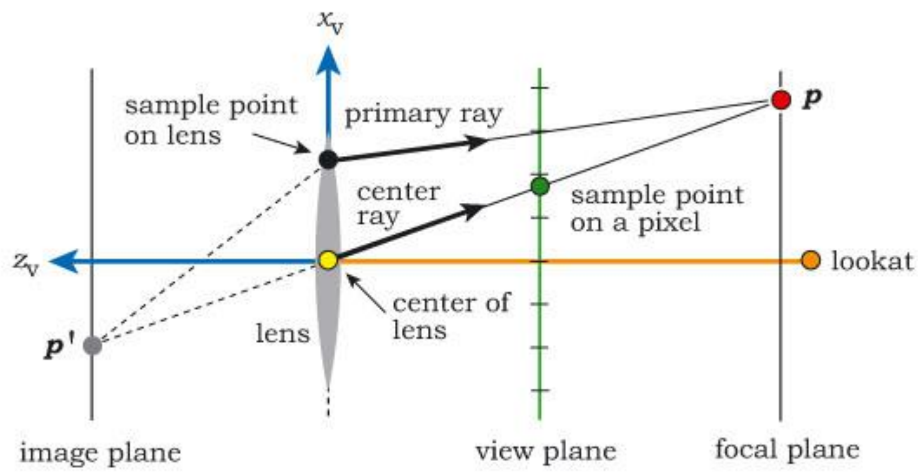
[Suffern]

Depth-of-Field

- is the range of distances to the lens in which the scene is in focus, i.e. the circle of confusion is smaller than the area of a pixel
- in cameras, the aperture is used to adapt the depth-of-field
  - narrow aperture $\Rightarrow$ large range of distances that are in focus
  - wide aperture $\Rightarrow$ small range of distances that are in focus

Simplified Model

- lens is modeled as disk placed at the eye position
- focal plane is defined by the user
- **p** is computed using a center ray from the center of a disk through a sample point in the view plane
- rays from the sampled disk through **p** are generated
- if these rays hit an object on the focal plane, the object is perfectly reconstructed

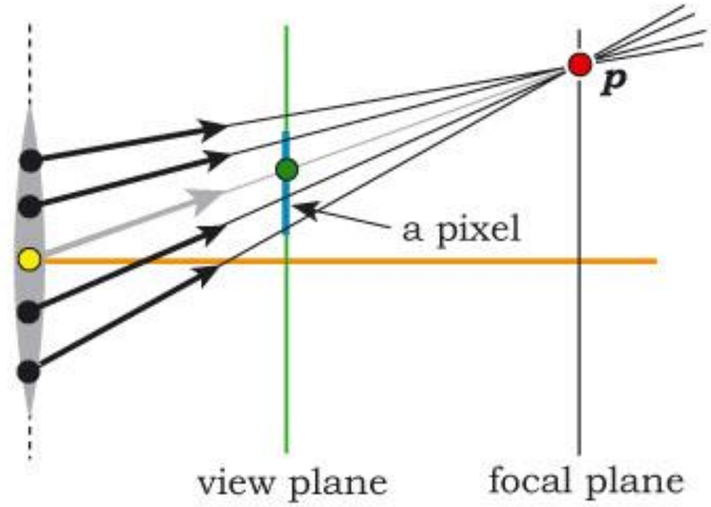


[Suffern]

UNI FREIBURG

Simplified Model

- center ray
  - computes **p**
  - does not return a radiance
- primary rays
  - start at a sample of the disk
  - into the direction of **p**
  - return a radiance that is associated with the pixel sample that has been used for the center ray
- size of the disk governs the blurring effect

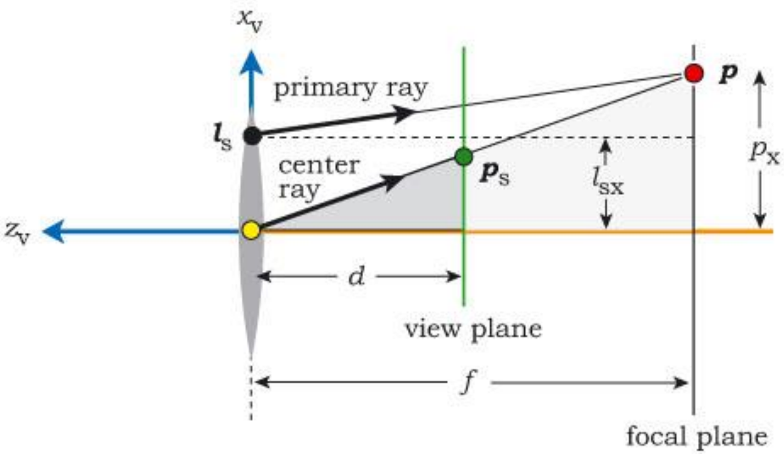


For a real lens, the black rays would be refracted. The rays would intersect at a point on the real image plane behind the lens.

[Suffern]

Implementation

- in view / camera space
  - a ray is computed for a lens sample $\mathbf{l}_s$ into the direction $\mathbf{d}=\mathbf{p}-\mathbf{l}_s$



$$\mathbf{p} = \left(\mathbf{p_s}_x \frac{f}{d}, \mathbf{p_s}_y \frac{f}{d}, -f\right)$$

$$\mathbf{d} = \left(\mathbf{p}_x - \mathbf{l_s}_x, \mathbf{p}_y - \mathbf{l_s}_y, -f\right)$$

- ray equation

$$\mathbf{r}(t) = \mathbf{l_s} + t\frac{\mathbf{d}}{\|\mathbf{d}\|}$$

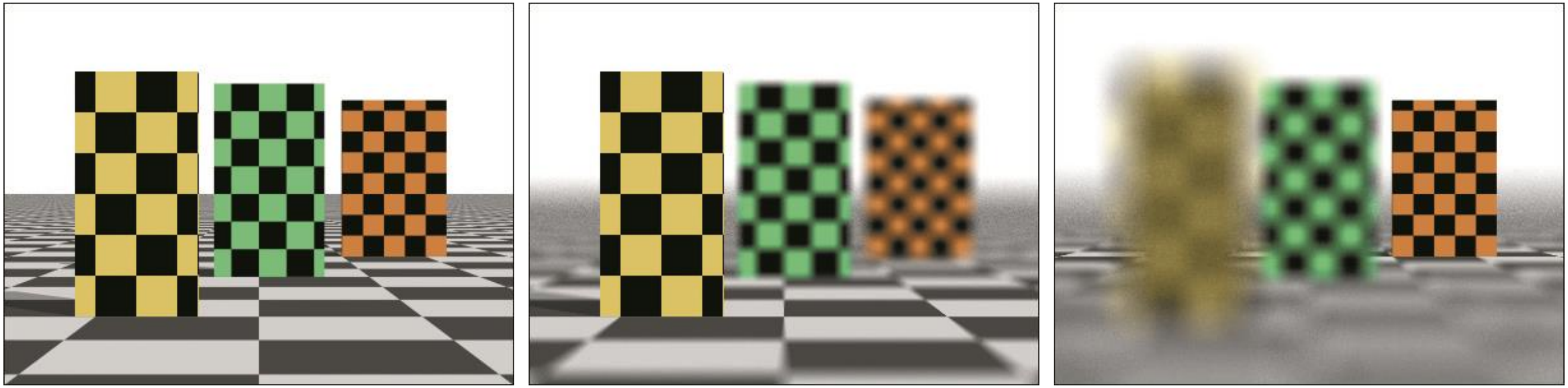
[Suffern]

Sampling

- sampling of the pixel area
- either
  - generate a center ray per pixel sample
  - sample the disk per center ray
- or
  - sample the disk
  - associate one disk sample with one pixel sample
  - i.e., use different center rays for all disk samples

Results

- 100 random pixel and disk samples
- one-to-one mapping of pixel and disk samples
- varying size of the disk



[Suffern]